실시간 영상압축과 복원시스템을 위한 DWT기반의 영상처리 프로세서의 VLSI 설계

정회원 서 영 호*, 종신회원 김 동 욱*

VLSI Design of DWT-based Image Processor for Real-Time Image Compression and Reconstruction System

Young-Ho Seo*, Dong-Wook Kim* *Regular Members*

요 약

본 논문에서는 이차원 이산 웨이블릿 변환을 이용한 실시간 영상 압축 및 복원 프로세서의 구조를 제안하고 ASIC(Application specific integrated circuit) 라이브러리를 이용하여 최소의 하드웨어로 구현하였다. 구현된 하드웨어에서 데이터 패스부는 웨이블릿 변환과 역변환을 수행하는 DWT 커널(Kernel)부, 양자화기 및 역양자화기, 허프만 엔코더 및 디코더, 웨이블릿 역변환 시 계수의 덧셈을 수행하는 덧셈기 및 버퍼, 그리고 입출력을 위한 인터페이스와 버퍼로 구성하였다. 제어부는 프로그래밍 레지스터와 명령어를 디코딩하여 제어 신호를 생성하는 주 제어부, 그리고 상태를 외부로 알리는 상태 레지스터로 구성된다. 프로그래밍 조건에 따라서 영상을 압축할 때의 출력은 웨이블릿 계수, 양자화 계수 혹은 양자화 인덱스, 그리고 허프만 코드 중에서 선택하여 발생할 수 있고 영상을 복원할 때의 출력은 허프만 디코딩 결과, 복원된 양자화 계수 그리고 복원된 웨이블릿 계수 중에서 선택하여 발생할 수 있다. 프로그래밍 레지스터는 총 16개로 구성되어 있는데 각각이 한번의 수직 혹은 수평 방향의 웨이블릿 변환을 수행할 수 있고 각각의 레지스터들이 차례대로 동작하기 때문에 4 레벨의 웨이브릿 변환을 한번의 프로그래밍으로 수행가능하다. 구현된 하드웨어는 Hynix 0.35m CMOS 공정의 합성 라이브러리를 가지고 Synopsys 합성 툴을 이용하여 게이트 레벨의 네트리스트(Netlist)를 추출하였고 이 네트리스트로부터 Vela 툴을 이용하여 타이밍 정보를 추출하였다. 추출된 네트리스트와 타이밍정보(sdf 파일)를 입력으로 하여 NC-Verilog를 이용하여 타이밍 시뮬레이션을 수행하여 구현된 회로를 검증하였다. 또한 Apollo 툴을 이용하여 PNR(Place and route) 및 레이아웃을 수행하였다. 구현된 회로는 약 5만 게이트의 적은 하드웨어 자원을 가지고 최대 80MHz에서 동작 가능하였다.

ABSTRACT

In this paper, we propose a VLSI structure of real-time image compression and reconstruction processor using 2-D discrete wavelet transform and implement into a hardware which use minimal hardware resource using ASIC library. In the implemented hardware, Data path part consists of the DWT kernel for the wavelet transform and inverse transform, quantizer/dequantizer, the huffman encoder/huffman decoder, the adder/buffer for the inverse wavelet transform, and the interface modules for input/output. Control part consists of the programming register, the controller which decodes the instructions and generates the control signals, and the status register for indicating the internal state into the external of circuit. According to the programming condition, the designed circuit has the various selective output formats which are wavelet coefficient, quantization coefficient or index, and Huffman code in image compression mode, and Huffman decoding result, reconstructed quantization coefficient, and reconstructed wavelet coefficient in image reconstructed mode. The programming register has 16 stages and one instruction can be used for a horizontal(or vertical) filtering in a level. Since each register automatically operated in the right order, 4-level discrete wavelet transform can be executed by a programming. We synthesized the designed circuit with synthesis library of Hynix 0.35um CMOS fabrication using the synthesis tool, Synopsys and extracted the gate-level netlist. From the netlist, timing information was extracted using Vela tool. We executed the timing simulation with the extracted netlist and timing information using NC-Verilog tool. Also PNR and layout process was executed using Apollo tool. The Implemented hardware has about 50,000 gate sizes and stably operates in 80MHz clock frequency.

*광운대학교 전자재료공학과 Digital Design & Test Lab.(ddntlab.kw.ac.kr, design@kw.ac.kr)

논문번호 : 030191-0509, 접수일자 : 2003년 5월 9일

※본 논문은 정보통신부 정보통신연구진흥원에서 지원하고 있는 정보통신기초연구지원사업의 연구결과(과제번호 : 03-기초-0025)입니다.

Ⅰ. 서 론

지난 10여년 동안 정보 및 데이터의 저장매체나 전송기술은 큰 발전을 이루었다. 또한 유선 통신에서 나아가서 무선 기반의 통신을 위한 제반적 기술 및 기반 인프라에 대한 구축이 괄목할 만한 성장을 이루었다. 이와 함께 사용자들은 많은 데이터와 함축적인 정보를 담을 수 있는 영상이나 비디오 서비스와 같은 대용량의 정보와 우수한 서비스를 요구하고 있어 이러한 대용량의 데이터들의 효율적인 처리기술의 중요성이 계속적으로 높아지게 되었고 이에 대한 연구가 활발히 진행되어 왔다. 영상 정보를 효율적으로 처리하고자 하는 가장 대표적인 기술이 JPEG과 MPEG 및 H.26X의 표준들이며, 이들 표준들을 응용한 소프트웨어(Software, S/W) 혹은 하드웨어(Hardware, H/W) 제품들이 경쟁적으로 쏟아져 나오고 있다. JPEG 또는 MPEG은 이산 코사인 변환(Discrete Cosine Transform, DCT)을 기반으로 하는 기술로서 여러 각도의 기술적 진보에도 불구하고 블록효과라는 필연적인 단점을 갖고 있다. 이를 보완 및 대체하기 위한 기술이 최근 10여년간 연구되고 있는데 대표적인 것이 웨이블릿(wavelet)을 기반으로 하는 영상처리이다. DCT와는 다르게 이산 웨이블릿 변환(Discrete Wavelet Transform, DWT)은 블록효과를 제거할 수 있을 뿐 아니라 전체영상을 대상으로 인간의 시각에 따른 처리가 가능하여 JPEG2000[1]의 표준 변환으로 이미 지정되었다.

DWT를 하드웨어로 구현하는 연구는 Knowles[2]로부터 시작되었는데 이 연구는 1차원 DWT (1-Dimentional DWT, 1D DWT)를 타겟으로 하고 있으며 중간 연산결과를 다음 연산에 사용하기 위해 다수 또는 대형 다중화기를 사용하여 실제적인 H/W 구현으로는 적합하지 않았다. 이 연구는 Lewis[3]의 연구로 이어져 DWT를 수행하는데 필요한 곱셈기의 사용을 피하는 알고리즘을 제안하였으나, 너무 제한적인 DWT 필터에만 적용 가능하여 일반적인 영상들에 대한 범용성이 매우 약하다. 1D DWT를 수행하는데 있어서 메모리를 효과적으로 사용하고자 하는 연구도 진행되었다[4]. 2D DWT에 대한 연구 중 가장 많은 비중을 차지하는 부분이 DWT를 수행하는 순서나 DWT 연산방법을 변경하고자 하는 것이다. 먼저, 1차원 DWT를 2차원으로 확장하여 분리가 가능한(Separable) 2차원 DWT 방법과 행과 열의 변환이 분리가 불가능(Non-separable)한 방법[5][6]으로 나누어 볼 수 있다. 분리가 불가능한 알고리즘은 복잡성 때문에 많은 레지스터를 사용하여야 하고 스케줄링이 복잡하여 근본적으로 분리 가능한 방법들에 비해 많은 재원을 필요로 한다. 분리 가능한 방법 중에는 영상의 주기적인 확장을 통하여 배치(batch) 처리가 가능하도록 하는 연산방식[7]과, 저대역 통과 및 고대역 통과과정을 병렬로 수행하는 방법[8], 연산 스케줄 표(Computation schedule table)를 형성하여 연산순서를 결정하도록 하는 방법[9] 등이 제안되었다. 특히 Vishiwanath[10]는 systolic 어레이와 파이프라인 구조를 결합한 연산형태를 제안하였다. 이외에 DWT를 효과적으로 수행하도록 하는 노력은 필터뱅크를 형성하고 필요에 따라 필터의 종류 및 필터의 길이를 조절할 수 있도록 한 방법[11] 등 다방면에서 연구가 진행되고 있으며, DWT를 수행하는 H/W를 IP화 하고자 하는 노력[12]과 FPGA에 사상하는 방법[13]도 연구되고 있다.

본 논문에서는 최소의 하드웨어 자원을 사용하면서 실시간으로 영상을 압축 및 복원할 수 있는 웨이블릿 기반의 프로세서를 제안한다. 이 프로세서는 기존의 설계와 차별성 및 경쟁력을 위해서 4개의 MAC(Multiplier-accumulator)을 사용하여 필터링을 수행하는 구조를 갖고 있으며 사용의 용이성을 위해서 통계적이면서 이론적인 바탕으로 구성된 양자화기와 허프만 코더를 내장한다. 또한 차세대 압축 표준인 JPEG2000을 위한 실시간 하드웨어 및 코어로의 사용을 고려하여 많은 유연성을 부여하여 설계한다. 본 회로는 호스트 프로세서와 함께 2차원 웨이블릿 변환을 이용해서 영상을 압축 및 복원할 수 있는 코프로세서(Co-processor)이고 DVR 및 웹 카메라 등의 영상 압축/복원 시스템을 구축하는데 이용한다. 또한 단일 칩으로써 영상압축을 요구하는 시스템에 사용될 수도 있으며, IP(Intellectual Property)화 되어 SOC(System on a Chip)를 설계할 때도 사용될 수 있다[14].

2장에서는 전체적인 하드웨어 구조 및 사양에 대해서 설명하고 3장에서는 설계된 하드웨어에서 중요한 블록들에 대해 내부 하드웨어 구조와 동작에 대해서 설명한다. 또한 4장에서는 회로를 프로그래밍하고 동작시키는 방식에 대해서 설명하고 설계된 하드웨어 대한 결과를 5장에서 나타낸다. 마지막으로 6장에서 결론을 맺는다.

II. 전체적인 프로세서 구조 및 사양

본 논문에서는 이차원 이산 웨이블릿 변환을 이용한 실시간 영상 압축 및 복원 프로세서의 구조를 그림 1과 같이 제안하고 최소의 하드웨어로 구현한다. 이 프로세서의 데이터 패스부는 웨이블릿 변환과 역변환을 수행하는 DWT 커널(Kernel)부, 양자화기 및 역양자화기, 허프만엔코더 및 디코더, 웨이블릿 역변환 시 계수의 덧셈을 수행하는 덧셈기 및 버퍼, 그리고 입출력을 위한 인터페이스와 버퍼로 구성된다. 제어부는 크게 세 부분으로 나뉘는데 프로그래밍 레지스터와 명령어를 디코딩하여 제어 신호를 생성하는 주 제어부, 그리고 상태를 외부로 알리는 상태 레지스터로 구성된다. 프로그래밍 조건에 따라서 영상을 압축할 때의 출력은 웨이블릿 계수, 양자화 계수 혹은 양자화 인덱스, 그리고 허프만 코드 중에서 선택하여 발생할 수 있고 영상을 복원할 때의 출력은 허프만 디코딩 결과, 복원된 양자화 계수 그리고 복원된 웨이블릿 계수 중에서 선택하여 발생할 수 있다.

표 1에 프로세서에 대한 사양을 나타내었다. 먼저 영상의 압축과 복원을 위한 하드웨어가 모두 내장되어 있고 프로그래밍 조건에 따라서 영상의 크기를 조절하는데, 실제적으로 처리되는 화소 혹은 계수의 수를 프로그래밍 하여 영상의 크기를 맞춘다. 내장된 양자화기와 허프만 코더를 사용할 경우 약 30dB의 PSNR(Peak signal-to-noise ratio)과 약 40대 1의 압축율을 겨냥하였다. 한번의 프로그래밍으로 최대 4레벨의 웨이블릿 변환이 가능하고 33MHz에서 동작시킬 경우 66 필드를 처리하도록 하여 실시간 동작이 가능하도록 하였다. Daubechies (9,7) 필터를 Booth 인코딩된 형태로 내장하여 필터링을 수행하고 내부적으로 웨이블릿 계수는 16비트, 필터 계수는 12비트의 수체계를 사용한다. 전반적인 동작은 프로그래밍에 의해서 구성되는데 프로그래밍은 영상 데이터를 입력받는 데이터 입력 포트를 그대로 사용한다. 따라서 데이터의 입력과 프로그래밍은 동시간에 일어날 수 없는 구조를 가진다. 영상의 크기와 웨이블릿 필터링 레벨을 비롯한 전반적인 압축 및 복원 과정은 프로그래밍에 의해서 결정된다. 프로그래밍 레지스터는 총 16개로 구성되어 있는데 각각이 한번의 수직 혹은 수평 방향의 웨이블릿 변환을 수행할 수 있고 각각의 레지스터들이 차례대로 동작하기 때문에 4레벨의 웨이브릿 변환을 한번의 프로그래밍으로 수행할 수 있도록 하였다.

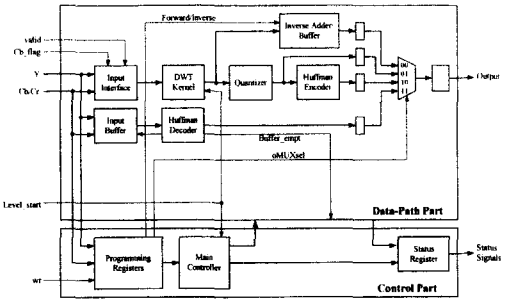


그림 1. 프로세서 전체의 블록도
Fig 1. Global block diagram of the processor

III. 영상 압축/복원 프로세서의 하드웨어 구조

본 장에서는 앞장에서 언급된 프로세서 구조 중 일부 중요한 블록에 대해서 구조와 동작을 설명한다

3-1. DWT Kernel의 구조

표 1. 프로세서의 설계 사양
Table 1. Design specification of the processor

Catagory	Specification
Transform	Forward/Inverse
Image size	User defined
Image form	NTSC YCbCr(4:2:2)
Compression rate	over 40:1
PSNR	about 30dB
Number system	Pixel:16-bit(9,7),Filter:10-bit(2,10)
Operation performance	67field/sec(33 frame/sec)
The number of level	Max. 4 level
DWT filter	Daubechies (9,7) filter
Quantizer	linear fixed, exception index
Entropy coding	Huffman coding
Programming Register	16

영상압축기에서 핵심부분은 2D DWT를 수행하는 DWT 커널이다. 커널은 곱셈과 누적덧셈을 수행하는 연산기(Multiplier and Accumulator, MAC)로 구성되는데, 33MHz의 기준 주파수를 사용하므로 1개의 MAC으로는 실시간 변환을 수행할 수 없고 기존 연구에서와 같이 많은 수의 곱셈기를 사용할 경우 H/W의 양이 과다해져서 기존의 연구와 차별성을 가지지 못하고 하드웨어의 양이 방대해진다. 따라서 본 논문에서는 최소의 H/W 자원을 사용하면서 실시간성을 가지는 4개의 MAC을 사용하는 커널구조로 설계하였다. 이 구조를 그림 2에 나타내었는데, 다중 쉬프터(Multi-Shifter)로 이루어진 프리-버퍼(pre-buffer), 내부 듀얼-포트 램(Dual-port RAM)으

로 구성된 RAM 체인, 32비트 CLA(Carry Look Ahead Adder)로 구성된 선-덧셈, 그리고 누적 기능을 가진 하이브리드 CSA 트리(Hybrid Carry Save Adder tree)와 Booth 곱셈기 및 CLA로 구성된 MAC 열의 구조를 갖고 있다.

Y, Cb, 그리고 Cr의 컬러 색차성분을 가지는 영상 데이터를 웨이블릿 필터링을 할 경우 세 개의 프리-버퍼는 각각 Y, Cb, 그리고 Cr 화소 성분을 위한 것으로 DWT 필터링의 동작 단계에 따라 해당 화소들을 입력받는다. 각 동작 상태에 따라 Y 혹은 Cb/Cr 성분이 프리-버퍼에 입력된 후에는 RAM 체인, 선-덧셈기, 부분곱 생성기(PP generator) 및 MUX, CSA 트리 그리고 최종 덧셈기까지 5단계의 파이프라인 단계를 거쳐서 하나의 화소에 대한 DWT 계수를 생성한다. 또한 하나의 DWT 계수를 만드는데 5번의 누적과정을 거치기 때문에 첫 DWT 계수가 출력되는데는 9 클록이 소요되고 다음 계수부터는 5 클록마다 하나씩 출력된다.

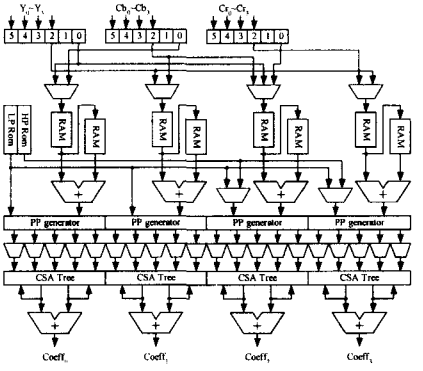


그림 2. DWT Kernel의 하드웨어 구조
Fig. 2. Hardware structure of DWT kernel

3-2. 허프만 코더 및 양자화기의 구조

데이터 압축은 양자화와 엔트로피 코딩(허프만 코딩)의 두 과정을 거쳐 이루어진다. 본 설계에서는 양자화 영역의 발생 빈도수로부터 허프만 코드를 발생시켜 양자화 인덱스에 대해 허프만 코드를 지정하였고 그에 대한 하드웨어를 설계하여 내장시켰다[15]. 양자화기는 선형 스칼라 양자화기로 구현된 양자화기이고 부대역에 따라 양자화기가 두 부분으로 나누어져 있으며, FIFO에서 정렬되어 양자화 인덱스, 예외계수 그리고 양자화 영역에 대한 정보를 출력한다. 양자화기 및 허프만 코더는 비교기와 ROM 테이블로 이루어져 있는데 비교기에 의해 ROM의 주소가 결정되면 ROM으로부터 허프만 코드와 유효 비트를 나타내는 비트 정보가 출력되고 이들과 함께 DWT 계수가 FIFO를 거쳐 출력된다. 5 클록마다 최대 MAC으로부터 4개의 출력이 발생하므로 이를 순차적으로 처리하기 위해 FIFO를 사용하여 MAC과 시간적인 완충작용을 수행한다.

3-3. 허프만 디코더 및 역양자화기의 구조

허프만 디코더는 역양자화기와 결합된 형태를 가지는데 그림 3에 보이는 것과 같이 허프만 디코딩 결과로 16비트의 양자화 계수(Coefficient[15:0])를 출력한다. 16비트 두 개의 입력 포트로부터 32비트 단위의 입력 데이터를 받고 코드경계 검출기(Delimiter detector)에 의해서 부대역 정보를 비롯하여 영상에 대한 데이터를 추출하고 직렬 쉬프터(HD shifter)를 통해 직렬 데이터를 발생시켜 허프만 디코딩을 수행한다. 디코딩을 하면서 예외 영역에 대한 정보가 검출되면 그 후의 4개의 직렬 데이터를 웨이블릿 계수로 변환하여 출력한다. 최저 주파수 대역의 데이터가 입력되는 것이 검출되면 허프만 디코딩을 거치지 않고 직렬 데이터들을 곧바로 웨이블릿 계수로 변환하여 출력한다.

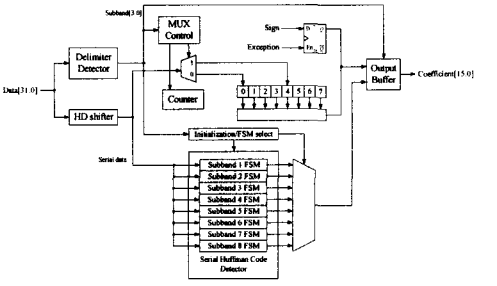


그림 3. 허프만 디코더와 역양자화기의 구조
Fig. 3. Structure of Huffman decoder and de-quantizer

3-4. 입력 인터페이스의 구조

그림 4에 나타난 입력 인터페이스는 외부로부터 데이터를 입력받기 위해 사용된다. 메모리를 이용할 경우를 위하여 일반적 SDRAM의 시동단계(Power-up sequence)를 수행하는 회로를 내장하였고 A/D 변환기를 직접 연결하여 사용할 경우를 위하여 I2C 제어기를 내장하고 있다. 본 논문에서는 Bt829b A/D 변환기를 가정하여 설계되었다. 또한 A/D 변환기로부터 메모리로 데이터를 직접 입력하기 위한 기

능도 갖추고 있지만 여기서는 그 출력에 대한 포트를 사용하지 않는 것으로 한다.

A/D 변환기를 사용할 경우 비디오 제어 신호 처리부(Video control signal process)에서 비디오 제어신호를 분석하여 유효한 영상 데이터(Y, Cb, Cr-data)만을 비디오 입력버퍼(Video input buffer)에 저장시켜 사용한다. 또한 외부 메모리를 사용할 경우, 일반적으로 메모리는 높은 클록 주파수를 가지는데 이를 위해 비디오 입력버퍼에 저장된 데이터를 고속 동작을 위한 메모리 입력버퍼(Memory input buffer)로 옮겨서 메모리에 입력한다.

3-5. 역변환 덧셈기/버퍼의 구조

영상의 복원과정에서 웨이블릿 필터링 수행 시 고주파 성분의 부대역과 저주파 성분의 부대역에 대한 필터링 결과를 더해야 복원된 웨이블릿 계수가 추출된다. 따라서 역변환을 위한 덧셈기와 그에 따른 시간적 완충이 요구되는데 이를 역변환 덧셈기/버퍼에서 수행하며, 이를 그림 5에 나타냈다. 영상 압축과정에서는 덧셈과정이 필요 없으므로 by-pass를 하게 된다.

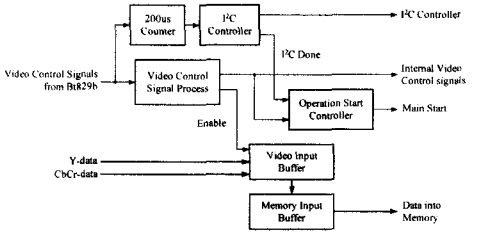


그림 4. 입력 인터페이스의 블록 다이어그램
Fig. 4. Block Diagram of input interface

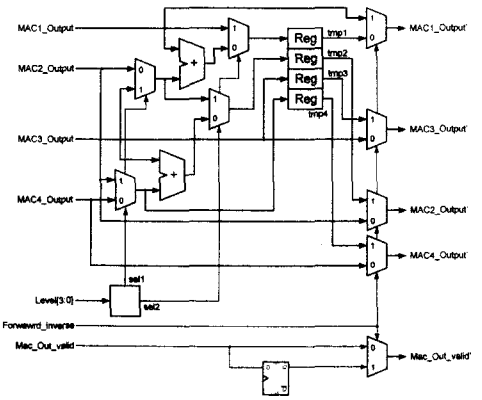


그림 5. 역변환 덧셈기/버퍼의 구조
Fig 5. Block diagram of Inverse adder/buffer

Ⅳ. 영상 압축/복원 프로세서의 동작

본 장에서는 H/W의 동작을 위한 프로그래밍 레지스터의 구조에 대해 살펴보고 프로그래밍에 따른 동작모드를 설명한다.

4-1. 프로그래밍 레지스터 및 상태 레지스터

프로그래밍 레지스터는 총 16개로 구성되어 있고 각각의 레지스터는 21비트로 구성된다. 즉, 프로그래밍 모드에서는 32비트의 입력 데이터 포트 중에서 21비트만을 사용한다. 하나의 레지스터 내용에 대한 동작이 완료되면 자동적으로 두 번째 레지스터 내용의 동작을 수행하게 되고 이러한 방식으로 총 16번의 독립적인 동작이 가능하다.

프로그래밍 레지스터의 내용은 표 2와 같다. 표 2의 1번에서 처리 화소를 위해 10비트를 프로그래밍이 가능할 수 있으므로 최대 150×1024×4개의 웨이블릿 계수를 한번의 프로그래밍으로 처리할 수 있다. 여기서 프로그래밍하는 10비트가 직접적인 처리 개수를 나타내는 것이 아니고 150번이 한번의 카운팅을 나타낸다. 또한 한번에 4개씩 웨이블릿 계수가 출력되기 때문에 4배에 해당하는 동작을 수행하게 된다. 그리고 몇 개의 레지스터에 프로그래밍을 했는지(Programming registers)와 웨이블릿 변환레벨은 얼마인가(Transformed level), 그리고 압축과정인지 복원과정인지(Compression/Reconstruction)를 프로그래밍한다. 각 과정에서 내부적으로 어느 모듈을 사용할지는 출력 다중화기를 선택함으로써 결정된다.

상태 레지스터는 현재의 칩 내부의 상태를 알려주는 것으로 아래와 같은 내용을 나타내고 총 8비트 크기의 레지스터를 사용하며, 표 3에 그 내용을 요약하였다. 먼저 상태 레지스터는 웨이블릿 변환 레벨의 시작(Level start signal)과 끝(Level end signal)을 외부로 알려주고 모든 출력 결과에 대해 유효한 값을 지시(Output valid)한다. 그리고 영상의 복원 과정에서 허프만 디코딩을 할 경우 디코딩을 위해 일정 데이터를 버퍼에 채워야 하는데 이러한 상태(Buffer empty in Huffman decoding)를 외부로 알려준다. 또한 모든 과정에서 현재 어느 부대역에 대한 처리가 진행되고 있는지(Processed subband state)를 알려줌으로써 외부의 호스트 프로세서로 하여금 다음 처리를 준비하게 한다.

표 2. 구현된 회로의 블록구성
Table 2. Block configuration of designed circuit

#	Content	The number of bits
1	The number of processed pixels	10
2	Programming registers	4
3	Output MUX selection	2
4	Transformed level	4
5	Compression/Reconstruction	1
Total		21

표 3. 상태 레지스터의 구성
Table 3. Configuration of status register

#	Content	# of bits
1	Level start signal	1
2	Output valid	1
3	Buffer empty in Huffman decoding	1
4	Level end signal	1
5	Processed subband state	4
Total		8

4-2. 프로그래밍에 따른 동작모드

프로그래밍에 따른 동작모드는 표 4와 같은데 크게 프로그래밍을 하는 모드와 영상을 압축하는 모드 그리고 영상을 복원하는 모드로 구성된다. 프로그래밍 모드의 경우 데이터 포트를 이용해서 프로그래밍 레지스터에 정보를 입력시키고 입력된 정보는 주 제어기에 의해 분석되어 제어신호를 발생시키는 기준이 된다. 영상을 압축하는 모드는 크게 세 가지 모드로 나누어지는데 각 모드는 표 2의 3번 조건에 의해서 결정된다. DWT 변환 출력 모드(Forward DWT filtering output mode)가 되면 입력 인터페이스를 통해 입력된 데이터는 DWT 커널을 통해 필터링을 수행하고 역변환 덧셈기/버퍼를 지나쳐 출력된다. 양자화 계수 출력모드(Quantization coefficient output mode)는 DWT 변환 출력모드에 양자화 과정을 추가하여 양자화 계수나 인덱스를 출력하고 허프만 코드 출력모드(Huffman code output mode)는 양자화 인덱스에 대해 확률적으로 지정된 허프만 코드를 출력한다. 영상을 복원하는 모드는 두 가지 모드로 나누어지고 각 모드는 역시 표 2의 3번 조건에 의해서 결정된다. 허프만 디코딩 출력모드(Huffman decoding output mode)는 32비트 단위의 허프만 코드를 버퍼의 크기만큼 요구하여 허프만 디코더를 통해서 양자화 계수로 복원하여 출력한다. 그리고 역 DWT 필터링 출력모드(Inverse DWT filtering output mode)는 복원된 양자화 계수를 받아들여 DWT 커널을 통해 역 DWT를 수행하고 상관성이 있는 계수끼리 역변환 덧셈기/버퍼를 통해 덧셈과정을 거친 후 출력한다.

표 4. 프로그래밍에 따른 동작모드
Table 4. Operation mode by programming

#	Mode	Detail mode	Operation sequence
1	Programming mode	Programming operation	Programming register → Main controller
2	Image compression mode	Forward DWT filtering output mode	Input interface → DWT kernel → Inverse adder/buffer (bypass mode) output
		Quantization coefficient output mode	Input interface → DWT kernel → Quantizer output
		Huffman code output mode	Input interface → DWT kernel → Quantizer → Huffman encoder output
3	Image reconstruction mode	Huffman decoding output mode	Input buffer → Huffman decoder → output
		Inverse DWT filtering output mode	Input interface → DWT kernel → Inverse adder/buffer → output

V. 설계 및 시뮬레이션

제안된 영상 압축/복원 프로세서는 VHDL (VHSIC Hardware Description Language)을 이용하여 하드웨어로 설계하였다[16]. 또한 설계된 하드웨어는 시스템내의 호스트 프로세서와 함께 실시간으로 영상압축 및 복원을 수행하도록 하였다. 설계는 VHDL 하향식(top-down) 설계 기법을 통해서 이루어졌으며 특정 구현 대상에 국한하지 않는 범용적인 설계를 이루고자 오직 IEEE 표준 라이브러리만을 사용하였다. 각각의 모듈들은 RTL(Registar Transfer Level)수준으로 설계되었고, 구조적 수준(structure-level)에서 서로 연결되었다. 설계된 하드웨어를 검증하기 위해 Synopsys의 Design Complier™로 논리 합성을 수행하였다[17]. 그림 6에 합성 및 시뮬레이션 절차를 나타내었다.

구현된 회로는 Hynix의 0.35um CMOS 라이브러리를 이용하여 Synopsys에서 합성되었는데 NAND를 한 개의 게이트로 환산할 경우 구현된 하드웨어는 약 5만 게이트의 자원을 사용한다. Synopsys에서 합성된 회로의 타이밍 시뮬레이션은 NC-Verilog에서 이루어졌고 타이밍 시뮬레이션 결과 약 80MHz의 클록 속도에서 안정적으로 동작함으로 확인하였다. 본 논문에서 설계된 회로는 컴퓨터와의 인터페이스 혹은 A/D 변환기와의 동작적인 연결성을 위해서 33Mhz

혹은 27Mhz의 동작조건을 만족하면 되기 때문에 타이밍 시뮬레이션을 통해 검증된 결과는 만족할 만하였다. 그림 7에 시뮬레이션 결과 중 가장 중요한 동작인 압축을 위한 웨이블릿 필터링 과정의 일부를 보였다. 그림에서 "wr"과 "en" 신호는 각각 프로그래밍 데이터와 영상 데이터에 대해 유효한 입력을 나타내고 "y"와 "c"는 데이터 포트를 나타낸다. "ini_start"와 "main_start"는 초기화 신호 및 동작시작 신호이고 "status"와 "dataout"은 각각 상태와 출력 신호이다. 그림에서 보는 것과 같이 "wr" 신호와 함께 데이터 포트("c")를 이용해서 칩의 동작을 결정하기 위한 프로그래밍 과정(Programming Operation)을 수행한다. 프로그래밍에 의해서 동작이 결정되면 "ini_start" 신호를 통해서 필터링을 위한 초기화 과정을 수행하고 데이터("Y", "C")와 "en" 신호를 이용해서 영상을 입력시키고 필터링을 수행(Image Input & Filtering Operation)한다. 그림 7의 시뮬레이션 결과에서 "en"이 '1'인 타이밍에서 "y"와 "c"의 각각 16비트 데이터를 입력받아 필터링을 수행하고 "status"신호를 '0x02'로 출력하여 유효함을 보이면서 "dataout"으로 출력을 발생시키는 것을 확인할 수 있다.

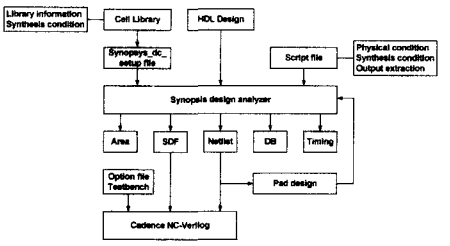


그림 6. 합성 및 시뮬레이션 절차
Fig 6. Synthesis and simulation procedure

구현된 DWT 커널은 [5]에서 제안된 구조와 가장 유사한데 차이점을 표 3에 나타냈다. 제안된 MAC의 구조는 내부적인 피드백 구조를 통해서 필터링 연산을 수행하기 때문에 [5]과 달리 필터 길이에 따른 곱셈기와 덧셈기의 숫자가 무관하다. 그림 2의 구조를 바탕으로 [5]에서 제안된 구조와 동일한 조건에서 비교할 경우의 결과를 표 5에 나타내었다. 여기서는 Daubechies의 (9,7) 필터를 사용하는 것으로 기준을 삼았다. 동일 데이터율과 레벨 처리방식이 적용될 경우에 본 논문에서 제안된 커널의 경우 총 12개의 곱셈기와 24개의 덧셈기를 사용하는 것에 해당하므로 [5]의 방식에 비해서 비교적 적은 H/W 자원을 사용한다. 모든 연산 메모리(working memory)를 외부 메모리를 사용하기 때문에 [5]과 달리 [6]에서 제안된 구조와 동일하게 N2의 storage가 요구되고 N2의 연산 시간(computing time)이 요구된다. 최소화된 H/W 구조를 보상하기 위한 입력 데이터 재 사용율을 높이는 RAM 체인의 제어가 복잡한 단점을 가진다. 그러나 [5]과 마찬가지로 최적의 하드웨어 사용율을 보인다.

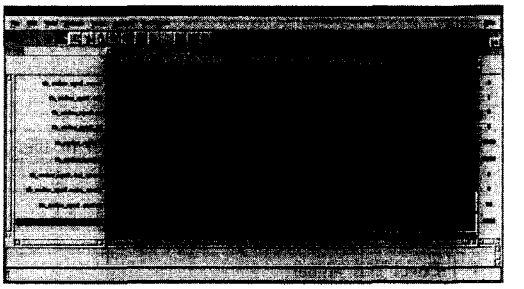


그림 7. 웨이블릿 필터링의 시뮬레이션 결과
Fig 7. Simulation result for wavelet filtering

표 5. 제안된 DWT 커널과 [5]의 성능 비교
Table 5. Performance comparison of the proposed DWT kernel and [5]

Architecture	Ours	[5]
Multiplier	12	36
Adder	24	36
Storage size	N2	N2/4+KN+K
Computing Time	N2	0.5N2~0.67N2
Control Complexity	Complex	Simple
Hardware Utilzation	100%	100%

합성된 회로에 입출력 포트를 위한 패드를 붙인 후 Apollo를 이용하여 P&R을 수행하였고 DRC 및 LVS 과정을 통해서 에러없이 결과를 추출하였다. 전체 chip 크기는 3860x3860mm이고 설계된 코어의 사용율은 약 70%에 해당한다. 그림 8에 Apollo를 이용하여 P&R을 수행한 결과를 나타내었다.

구현된 회로는 프로그래밍에 따라서 다양한 출력과 다양한 동작 방식이 가능하기 때문에 여러 시스템에 적용이 가능한데 그림 9에서는 본 논문을 통해서 구현된 하드웨어(P-DWT Processor)를 이용한 응용 시스템을 나타냈다. 먼저 그림 9의 (a)에서 보인 것과 같이 타일링(Tiling)과 ROI, 그리고 엔트로피 코딩을 위한 EBCOT 등을 부가하여 MJPEG2000의 구성이 가능하다. 또한 (b)에 보인 것과 같이 하나의 독립된 코덱으로써 DVR(Digital video recoder)등의 독립형 시스템을 구성하는데 영상 처리를 위한 핵심 부품으로 사용될 수 있다. 또한 최근 들어서 영상의

보안 및 보호를 위한 솔루션을 각광을 받고 있는 워터마킹을 위한 주파수 변환 도구로써 사용이 가능하고 영상의 보안통신을 위한 하드웨어 암호화 솔루션으로도 사용이 가능하다.

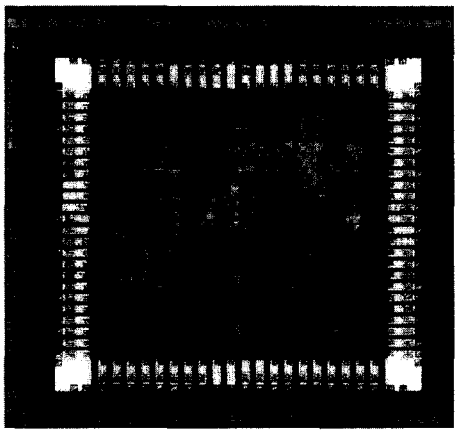


그림 8. Apollo를 이용한 P&R 결과
Fig 8. P&R result using Apollo

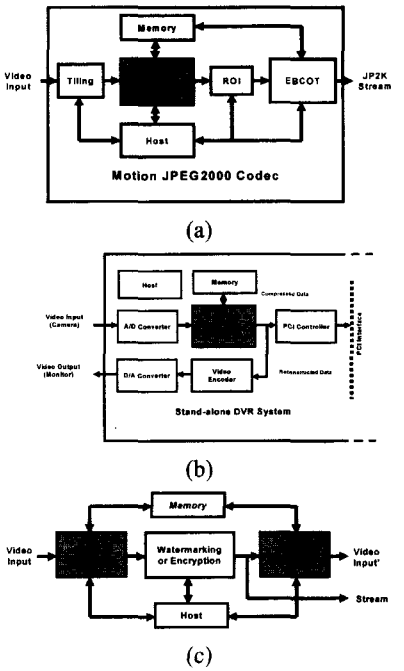


그림 9. 구현된 H/W를 이용한 응용 시스템 (a) MJPEG2000 코덱구현 (b) DVR 시스템의 구성 (c) 워터마킹 및 암호화 시스템의 구현
Fig. 9 Application system using the implemented H/W (a) Implementation of MJPEG2000 codec (b) configuration of DVR system (c) watermarking and encryption system

VI. 결론

본 논문에서는 최소의 하드웨어 자원을 사용하면서 실시간으로 영상을 압축 및 복원할 수 있는 웨이블릿 기반의 영상처리 프로세서 구조를 제안하고 하드웨어로 설계하였다. 설계된 하드웨어는 흑백 및 칼라 영상을 압축하고 복원할 수 있는 하드웨어를 모두 내장하고 있고 각 영상 성분에 대해 다양한 모드 동작이 가능하다. 기본적으로 선형 양자화기를 사용하면서 호스트 프로세서를 이용하여 가변적이고 적응적인 동작 구성이 가능하다. 각 동작을 간단한 프로그래밍을 통해서 계속적으로 변화시킬 수 있고, 상용 AD 변환기에 적합하게 필드단위로 동작하고 실시간 처리에 적합하게 NTSC 혹은 PAL의 필드 영상을 초당 66 필드 처리할 수 있어 실시간 응용에 범용적으로 사용이 가능하다. 또한 차세대 압축 표준인 JPEG2000을 위한 실시간 하드웨어 및 코어로의 사용을 고려하여 많은 유연성을 부여하여 설계하였다.

본 회로는 호스트 프로세서와 함께 2차원 웨이블릿 변환을 이용해서 영상을 압축 및 복원할 수 있는 코프로세서이고 DVR 및 웹 카메라 등의 영상 압축/복원 시스템을 구축하는데 이용한다. 또한 단일 칩으로써 영상압축을 요구하는 시스템에 사용될 수도 있으나 IP(Intellectual Property)화 되어 SOC(System on a Chip)의 설계에도 사용할 수 있다. 현재 국제표준으로 제정된 JPEG2000의 경우 웨이블릿 변환을 주 변환기법으로 사용하고 있고 이를 위한 다양한 제품이 개발되고 있는 시점에서 Motion JPEG2000을 위한 코어로써 중요한 역할을 할 수 있을 것으로 기대된다.

참 고 문 헌

[1] Martin Boliek, et al., JPEG 2000 Part I Final Draft International Standard, *ISO/IEC JTC1/SC29 WG1*, 24 Aug. 2000.

[2] G. Knowles, "VLSI Architectures for the Discrete Wavelet Transform", *IEEE Electronic Letters*, Vol. 26, No. 15, pp. 1184-1185, July 1990.

[3] A. S. Lewis and G. Knowles, "VLSI Architecture for 2-D Daubechies Wavelet Transform without Multipliers", *IEEE Electronic Letters*,

Vol. 27, No. 2, pp. 171-173, Jan. 1991.

[4] Jose Fridman and Elias S. Manolakos, "Distributed Memory and Control VLSI Architectures for 1-D Discrete Wavelet Transform", *IEEE Workshop on Signal Processing Systems*, pp. 388-397, 1994.

[5] Po-Cheng, Wu and Liang-Gee Chen, "An Efficient Architecture for Two-Dimensional Discrete Wavelet Transform", *IEEE Trans on Circuits and Systems for Video Tech.*, vol. 11, no. 4, April 2001

[6] C. Chakrabarti and M. Vishwanath, "Architectures for wavelet transforms: A survey," *J. VLSI Signal Processing*, vol. 14, pp. 171-192, 1996.

[7] Trieu-Kien Truong, et al., "A New Architecture for the 2-D Discrete Wavelet Transform", *IEEE Int'l Conf. of Communications Computers and Signal Processing*, pp. 481-484, 1997.

[8] Chu Yu and Sao-Jie Chen, "Design of an Efficient VLSI Architecture for 2-D Discrete Wavelet Transform", *IEEE Trans. on Consumer Electronics*, Vol. 45, No. 1, pp. 135-140, Feb. 1999.

[9] Ming-Hwa Sheu, Ming-Der Shieh and Sheng-Wet Liu, "A VLSI Architecture Design with Lower Hardware Cost and Less Memory for Separable 2-D Discrete Wavelet Transform", *IEEE ISCAS'98*, Vol. 5, pp. 457-460, 1998.

[10] Mohan Vishiwanath, Robert Michael and Mary Jane Irwin, "BSLI Architecture for the Discrete Wavelet Transform", *IEEE Trans. on Circuits and Systems-II: Analog and Digital Signal Processing*, Vol. 42, No. 5, pp. 305-316, May 1995.

[11] Jijin Chen and Magdy A. Bayoumi, "A Scalable Systolic Array Architecture for 2-D Discrete Wavelet Transforms", *IEEE Proc. of Midwest Symp. on Circuits and Systems*, Vol. 2, pp. 303-312, 1996.

[12] Shahid Masud and John V. McCanny, "Wavelet Packet Transform for System-on-Chip Application", *IEEE Proc. on ICASSP*, Vol. 6, pp. 3287-3290, 2000.

[13] Ali M. Reza and Robert D. Turney, "FPGA Implementation of 2D Wavelet Transform", *IEEE Conf. of Signals, Systems and Computers*, pp. 584-588, 1999.

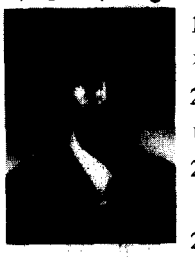
[14] Michael Keating and Pierre Bricaud, Reuse Methodology Manual, *Kluwer Academic Publishers*, 1999.

[15] Allen Gersho and Robert M. Gray, "Vector Quantization and Signal Compression", *Kluwer Academic Publishers*, 1992.

[16] I.S 1076-1993, IEEE Standard VHDL Language Reference Manual, *IEEE*, 1993.

[17] Pran Kurup and Taher Abbasi, Logic Synthesis Using Synopsys, *Kluwer Academic Publishers*, 1997.

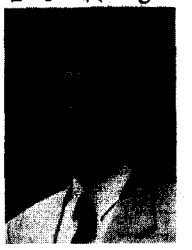
서 영 호(Young-Ho Seo) 정회원



1999년 2월 : 광운대학교 전자재료공학과 졸업(공학사).
2001년 2월 : 광운대학교 대학원졸업(공학석사).
2000년 3월~2001년 12월 : 인티스닷컴(주) 연구원.
2001년 3월~현재 : 광운대학교 전자재료공학과 박사과정.
2003년 6월~현재 : 한국전기연구원 연구원

<주관심분야> Image Processing/Compression, 워터마킹, 암호학, FPGA/ASIC 설계

김 동 욱(Dong-Wook Kim) 종신회원



1983년 2월 : 한양대학교 전자공학과 졸업(공학사).
1985년 2월 : 한양대학교 대학원 졸업(공학석사).
1991년 9월 : Georgia공과대학 전기공학과 졸업(공학박사).
1992년 3월~현재 : 광운대학교 전자재료공학과 정교수. 광운대학교 신기술 연구소 연구원.
2000년 3월~2001년 12월 : 인티스닷컴(주) 연구원.

<주관심분야> 디지털 VLSI Testability, VLSI CAD, DSP 설계, Wireless Communication